REMARKS

Claims 1-25 are pending. Claims 1, 15, 22 and 24 are independent claims.

Claims 1-5, 8, 11, 15-20 and 22-25 are rejected. Claims 6, 7, 9, 10, 12-14 and 21 are objected to.

The examiner uses Moller to reject claims 1, 15, 16, 20, 22 and 24 as having been anticipated.

Claim 1 recites "directing the processor having a plurality of microengines to swap a currently running context in a specified microengine out to memory to let another context execute in that microengine and cause a different context and associated program counter to be selected."

Moller neither describes nor suggests this quoted claim feature. Moller discloses programmable digital devices that use microprogramming, i.e., "...to a microprogram sequence controller that is readily microprogrammed through the use of structured microprogramming constructs and the availability of software and hardware debugging facilities, and that executes microinstructions rapidly and efficiently." (Col. 1, lines 15-21) Moller discloses nothing more than a traditional swap instruction for microinstructions in a traditional processor:

> Execution of the SWAP microinstruction causes CONTROL 100 to generate a CMUXCTL signal which is applied to C-MUX 118 causing it to pass the top of LIFO stack 136 contents on signal lines 140 to the down counter 120. At the same CONTROL 100 generates a STKMUXCTL signal which applied to STACK MUX 132 causes the STACK MUX to select the signal on lines 122 from the down counter 120 to be read onto the top of stack 136 upon receipt of the WE signal by stack 136 as subsequently generated by CONTROL 100. CONTROL 100 does not, under these circumstances, generate an UP signal so that the current top of stack contents are overwritten, having been loaded into down counter 120, so that the iteration count for the invoking loop replaces the iteration count for the invoked loop. (Col. 30, lines 39-54)

This is very different from directing the processor having a plurality of microengines to swap a currently running context in a specified microengine out to memory to let another context execute in that microengine and cause a different context and associated program counter to be selected. Accordingly, claim 1 is not anticipated by Moller.

Claims 15, 22 and 24 recite "performing a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter," or similar language. Moller neither describes nor suggests swapping in a multithreaded processor in conjunction with state. Accordingly, claims 15, 22 and 24 are not anticipated by Moller.

The examiner uses Moller and Adkins to reject claims 2, 3, 8, 17, 23 and 25 as having been obvious.

Claims 1, 17, 22 and 24 are not rendered obvious by Moller and Adkins. Moller was discussed above. Adkins fails to teach or suggest directing the processor having a plurality of microengines to swap a currently running context in a specified microengine out to memory to let another context execute in that microengine and cause a different context and associated program counter to be selected, or performing a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter. Adkins teaches:

> (A) serial communications adapter provides an interface to physical communications ports. A scheduler executing on the adapter schedules tasks at different priority levels, so that time-critical tasks are performed quickly enough to prevent data loss. Data to be transmitted or received through a communications port is stored in buffers on the adapter, and data and command communications between the adapter and the host system are preferably performed over a DMA channel. (Col. 1, lines 59-68)

No combination of Moller and Adkins teaches or suggests these independent claim features. Claims 2, 3, 8, 17, 23 and 25 depend upon, and add further limitations to, claims 1, 17, 22 and 24. Accordingly, claims 2, 3, 8, 17, 23 and 25 are not rendered obvious by Moller and Adkins.

The examiner uses Moller, Adkins and Angle to reject claims 4 and 18 as having been obvious.

Claims 1 and 15 are not rendered obvious by Moller, Adkins and Angle. Moller and Adkins were discussed above. Angle does not teach or suggest directing the processor having a plurality of microengines to swap a currently running context in a specified microengine out to memory to let another context execute in that microengine and cause a different context and associated program counter to be selected, or performing a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter. Angle merely teaches an improved packet streaming engine that manipulates data contained within a stream of packets using a delayed packet mechanism. "The delayed replace mechanism allows for packet streaming where embedded fields within a data packet may be modified dependent upon fields following them within the data packet." (see Abstract)

No combination of Moller, Adkins and Angle teaches or suggests this independent claim feature. Claims 4 and 18 depend upon, and add further limitations to, claims 1 and 15. Accordingly, claims 4 and 18 are not rendered obvious by Moller, Adkins and Angel.

The examiner uses Moller, Adkins and Manning to reject claims 5 and 19 as having been obvious.

Claims 1 and 15 are not rendered obvious by Moller, Adkins and Manning. Moller and Adkins were discussed above. Manning does not teach or suggest directing the processor having a plurality of microengines to swap a currently running context in a specified microengine out to memory to let another context execute in that microengine and cause a different context and associated program counter to be selected, or performing a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter, or performing a swapping operation to cause a different context and associated program counter to be selected in accordance with the value of the specified parameter. Manning merely teaches an integrated circuit memory device with a standard DRAM pinout that is designed for high speed data access and for compatibility with existing memory systems. (see col. 3, lines 14-16)

No combination of Moller, Adkins and Manning teaches or suggests these independent claim features. Claims 5 and 19 depend upon, and add further limitations to, claims 1 and 15. Accordingly, claims 5 and 19 are not rendered obvious by Moller, Adkins and Manning.

The examiner uses Moller, Adkins and Turner to reject claim 11 as having been obvious.

Claim 1 is not rendered obvious by Moller, Adkins and Turner. Moller and Adkins were discussed above. Turner does not teach or suggest directing the processor having a plurality of microengines to swap a currently running context in a specified microengine out to memory to let another context execute in that microengine and cause a different context and associated program counter to be selected.

Turner teaches "...non-executing threads exist in one of three states: Ready (40), Suspended (41), and Suspended-Waiting (42). This minimized set of possible thread states further enables efficiency of the embedded processing system. Threads are initially created in the Ready state (40). In the preferred embodiment, Ready threads are linked members of a Ready list, and are queued and awaiting for some combination of synchronous and asynchronous preemption events to advance in the Ready list and run." (col. 5, lines 56-64) Further, Turner merely uses techniques well known in the art:

> Control of the processor execution can be transferred from one thread to another by synchronous or asynchronous preemption. This concept is common in the art. Synchronous preemption is based on an event, such as receipt of a hardware interrupt signal, or the expiration of a hardware timer. (Col. 6, lines 38-43)

No combination of Moller, Adkins and Turner teaches or suggests this independent claim feature. Claim 11 depends upon, and adds further limitations to, claim 1. Accordingly, claim 11 is not rendered obvious by Moller, Adkins and Turner.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue or comment does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this

paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: January 11, 2005

Kenneth F. Kozik
Reg. No. 36,572

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21008180.doc